Outline of P1451.5 Draft Specification

June 7, 2004

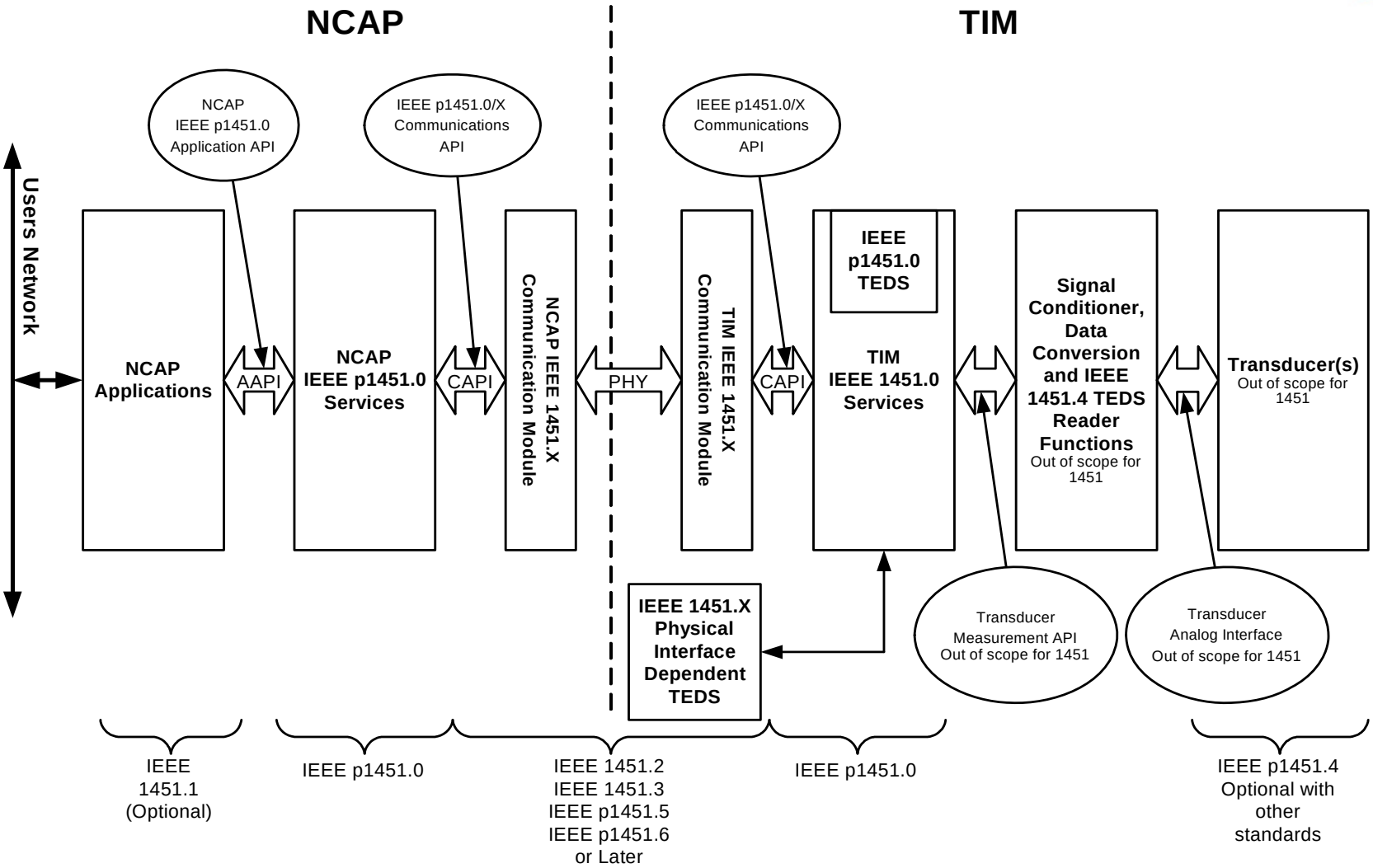
Ryon Coleman
Senior Systems Engineer
3eTI

rcoleman@3eti.com

Agenda

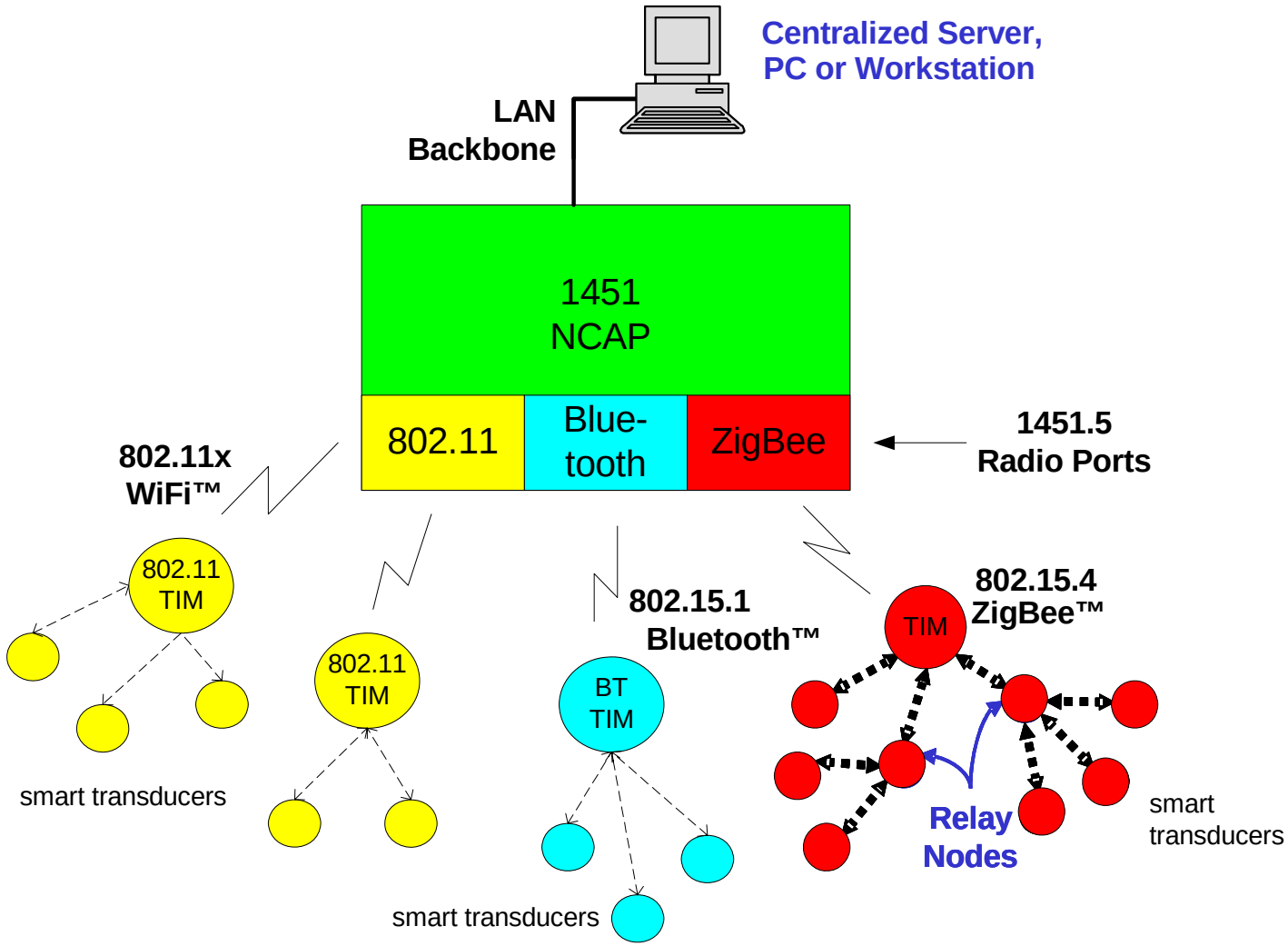
1. **Overview of the 1451 Reference Model**
2. **Envisioned Wireless Structure**
3. **Relationship Between 1451.5 and 1451.0**
4. **1451.5 Use of Core 1451.0 TEDS**
5. **802.11 Envisioned PHY TEDS**
6. **ZigBee Envisioned PHY TEDS**
7. **Way Ahead**
8. **Envisioned API with 1451.0**
9. **Discussion**

Overview of the 1451 Reference Model

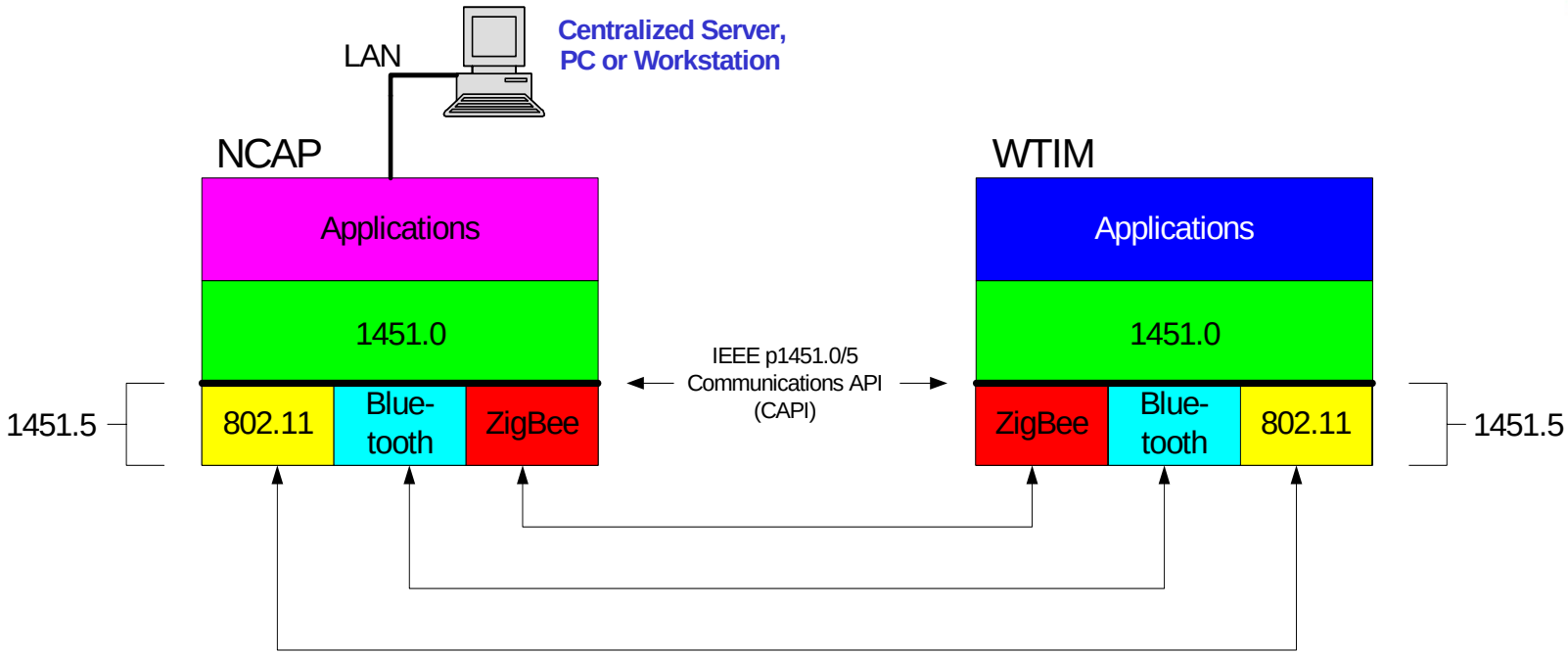


* Diagram from Lee Eccles, Boeing

Envisioned Wireless Structure



Relationship Between 1451.5 and 1451.0



1451.5 Draft will focus on:

- IEEE 1451.0/5 Communications API
- IEEE 1451.5 PHY TEDS: 802.11/BT/ZigBee
- Any commands that terminate in 1451.5 Layers

P1451.5 Use of Core P1451.0 TEDS

- **Meta TEDS**
  - Store worst-case timing parameters used by the NCAP to set time-out values to determine when the TIM is not responding. The remainder of this TEDS describes the relationships between the Transducer Channels that exist within the TIM.
- **Transducer Channel TEDS**
  - Provides detailed information about a specific transducer: what physical parameter is being measured or controlled, the range over which the Transducer Channel operates, the characteristics of the digital I/O, the operational mode(s) of the unit and the timing information.
- **User's Transducer Name TEDS**
  - (legacy "Commissioning" TEDS)
  - This TEDS is intended to provide a place for the user of the transducer to store the name by which the system will know the transducer. The structure of this TEDS is recommended in the standard but since the contents are user defined it cannot be required. All the manufacturer of the TIM needs to do is to provide the blank non-volatile memory that the user can write using the standard TEDS access methods.
- **1451.5-Specific PHY TEDS**
  - **802.11 PHY TEDS**
  - **Bluetooth PHY TEDS**
  - **ZigBee PHY TEDS**

802.11 Envisioned PHY TEDS

-- 802.11 MAC Attributes
-- DEFINED AS "The MAC object class provides the necessary support
-- for the access control, generation, and verification of frame check
-- sequences, and proper delivery of valid data to upper layers.";

- dot11mac OBJECT IDENTIFIER ::= {ieee802dot11 2}
  -- MAC GROUPS
  - -- **dot11OperationTable** ::= {dot11mac 1}
  - -- **dot11CountersTable** ::= {dot11mac 2}
  - -- **dot11GroupAddressesTable** ::= {dot11mac 3}

-- 802.11 PHY Attributes
-- DEFINED AS "The PHY object class provides the necessary support
-- for required PHY operational information that may vary from PHY
-- to PHY and from STA to STA to be communicated to upper layers."

- dot11phy OBJECT IDENTIFIER ::= {ieee802dot11 4}
  -- phy GROUPS
  - -- **dot11PhyOperationTable** ::= {dot11phy 1}
  - -- **dot11PhyAntennaTable** ::= {dot11phy 2}
  - -- **dot11PhyTxPowerTable** ::= {dot11phy 3}
  - -- **dot11PhyFHSSTable** ::= {dot11phy 4}
  - -- **dot11PhyDSSSTable** ::= {dot11phy 5}
  - -- **dot11PhyIRTable** ::= {dot11phy 6}
  - -- **dot11RegDomainsSupportedTable** ::= {dot11phy 7}
  - -- **dot11AntennasListTable** ::= {dot11phy 8}
  - -- **dot11SupportedDataRatesTxTable** ::= {dot11phy 9}
  - -- **dot11SupportedDataRatesRxTable** ::= {dot11phy 10}

ZigBee Envisioned PHY TEDS

- **802.15.4-ZigBee MAC attributes**
  - **ZBmacAckWaitDuration**
  - **ZBmacAssociationPermit**
  - **ZBmacBattLifeExt**
  - **ZBmacBeaconOrder**
  - **ZBmacBeaconTxTime**
  - **ZBmacGTSPermit**
  - **ZBmacMaxCSMABackoffs**
  - **ZBmacMinBE**
  - **ZBmacRxOnWhenIdle**
  - **ZBmacSuperframeOrder**

- **802.15.4-ZigBee PHY attributes**
  - **ZBphyChannelsSupported**
  - **ZBphyTransmitPower**
  - **ZBphyCCAMode**

* Provided by Ken Cornett, Motorola

Way Ahead

- Rigorously define IEEE 1451.0/5 Communications API (CAPI)
  - Pub/sub & SetQoS need work
- Fully Define 1451.5 PHY TEDS
  - 802.11
  - Bluetooth
  - ZigBee
- Establish “thin” 1451.5 convergence layer above 802.11, BT, ZigBee
  - Only required if Dot0 “staircase” approach abandoned
- Plan is to make headway on these categories at Sensors Expo, complete draft by July 31, 2004.

Envisioned API with Dot0

- **Open**
  - Format: OPEN (local port, foreign socket, active/passive [, timeout] [, precedence] [, security/compartment] [, options]) -> local connection name
- **Send**
  - Format: SEND (local connection name, buffer address, byte count, PUSH flag, URGENT flag [,timeout])
- **Receive**
  - Format: RECEIVE (local connection name, buffer address, byte count) -> byte count, urgent flag, push flag
- **Close**
  - Format: CLOSE (local connection name)
- **Status**
  - Format: STATUS (local connection name) -> status data
- **Abort**
  - Format: ABORT (local connection name)

Discussion…